Express Mail Label No. EL952274035

[0001] TURBO CODE ENCODER HAVING AN IMPROVED INTERLEAVER

[0002] Cross Reference to Related Applications

[0003] This application is a continuation of Application No. 09/806,202, filed March 27, 2001 which is a Section 371 National Phase of PCT/US99/24066, filed October 12, 1999 which claims priority from Provisional Application No. 60/112,318, filed December 14, 1998 and Provisional Application No. 60/104,040, filed October 13, 1998.

[0004] BACKGROUND OF THE INVENTION

[0005] Field of the Invention

[0006] The present invention relates generally to processes that create time diversity in systems with high processing gain. More specifically, the invention relates to a system and method of turbo code interleaving mapping where the number of tail bits required to flush the storage registers of each constituent encoder to an all-zero state are reduced.

[0007] Description of the Prior Art

[0008] In many types of data communication systems, whether voice or non-voice, signal diversity or redundancy when transmitting information is shown to improve performance without compromising other aspects of the data transmission system. Two techniques that add time diversity are known as interleaving and forward error-correcting (FEC) coding.

[0009] The process of interleaving is where the input data sequence is permuted or reordered into another sequence. For example:

$$(O\ 1\ 2\ 3\ 4\ 5\ 6\ 7) \xleftarrow{I_N} (3\ 0\ 6\ 7\ 1\ 5\ 2\ 4)$$

where the mathematical operator I_N *[J]* transposes the original position of each bit or symbol of a finite input sequence to a new position *J* by operation of the interleaver I_N. This reordering process that achieves time diversity is called interleaving and can be performed in a number of ways. Two methods of typical interleaving are known as block and random interleaving.

[0010] At the transmission destination, the signal is again reordered, putting the data sequence back in the original order. The inverse process is called deinterleaving.

[0011] The most recent advance in coding techniques which exhibit the best performance are turbo codes. A variety of turbo code interleaver designs exist and require less complexity when decoding. The three most popular are: 1) block interleavers; 2) pseudo-random interleavers; and 3) S-random interleavers.

[0012] The best performing interleavers are the S-random interleavers. The S-random interleavers exploit the property of not mapping neighbor positions within a certain sequence length, to neighbor positions exhibiting the same length. This makes the sequence length as large as possible. All interleaver designs require a specific set of rules setting forth input sequence size and permutation.

[0013] In conjunction with interleaving, FEC coding improves performance for signals that are coherently demodulated. FEC coding adds additional redundancy in the original data sequence. In communication systems that communicate over a spread spectrum air interface, redundancy is already present in the shared spectral transmission channel. A FEC encoder is a finite-state machine that relies upon nodes or states and delay registers. The predetermined transitions between the registers define a path from which a given data input may produce an output. A common way to illustrate the encoding and decoding technique for the convolutionally encoded data is the use of a trellis diagram which is known to those familiar with this art. A trellis diagram is an infinite replication of a state machine diagram and is shown in Figure 1.

[0014] The decoding is typically performed using a maximum likelihood algorithm which relies upon the trellis structure and the path state or metric for each level and each selected node or state. Any code word of a convolutional code corresponds to the symbols along a path in the trellis diagram. At each state and at each level of the trellis an add-compare-select operation is performed to select the best path and state. The trellis is assembled over many received symbols. After a predefined number of symbols have been accumulated, the determination finds the trellis path with the smallest error. The final decision on all bits in the trellis is made via the encoders by forcing the encoder to return to an initial all-zero state. This is achieved by inserting zero tail bits at the end of the finite bit stream after encoding. This process is referred to as "tailing off."

[0015] A process known as "chaining back" is performed starting at the last node, tracing the decision path back from the last decision to the first. This method of decoding determines which symbol was originally sent. The trellis structure introduces redundancy and accumulates past history.

[0016] A prior art turbo encoder is shown in Figure 2. The encoder comprises first and second systematic recursive convolutional code (RCS) encoders coupled in parallel with a turbo code interleaver coupled prior to the second recursive convolutional encoder. The two recursive convolutional codes used in each encoder are known as the constituent codes. The first encoder reorders the input information bits $\vec{x}_N$ in their original order while the second encoder reorders the input bits as permuted by the turbo code interleaver $\vec{x}^{I}{}_N$. The input information sequence $\vec{x}_N$ is always transmitted through a channel. In dependence upon the data transmission rate, the outputs from both encoders may be "punctured" before transmission $\vec{y}_N$. Puncturing is a process where alternate outputs of the lower taps (first and second encoders $\vec{p}^{1}{}_N, \vec{p}^{2}{}_N$) are deleted from the output. This process establishes a code rate.

[0017] The turbo code interleaver is a scrambler defined by a permutation of the sequence length with no repetitions. A complete sequence is input into the interleaver and output in a predefined order.

[0018] A prior art tailing off process is shown and described in Figures 3 and 4. The tail bits for each encoder are obtained from register feedback from each respective encoder as shown in Figure 3. Since the register contents of each constituent encoder are different at the beginning of the tailing off operation, each encoder must be flushed separately. As described in Figure 4, each encoder (in Figure 3) is flushed independently and exclusive of each other after the information bits have been encoded. Each encoder derives and receives its own tail bits. Therefore, if m equals the number of states or register memory of an encoder, m tail bits are required for one encoder and 2m are required for both encoders.

[0019] A prior art turbo code decoder is shown in Figure 5. On receiving the demodulated soft value signal $\bar{y}_N$, the soft-decision information for the systematic (information) and parity bits $\vec{p}^1{}_N$ from the first constituent encoder are input to a first constituent decoder. The first constituent decoder generates updated, soft-decision likelihood values $\vec{L}_{e_1}(\vec{x}_N)$ for the information bits that are input along with the information bits to a decoder interleaver. The input to a second constituent decoder includes the interleaved soft-valued sequences $\vec{x}^I{}_N$ *and* $\vec{L}^I{}_{e1}(\vec{x}_N)$ and the parity bits $\vec{p}^2{}_N$ from the second constituent encoder. The output of the second decoder improves on the soft-decision likelihood values derived from the output from the first constituent decoder and is fed back to the first constituent decoder after reordering in accordance with the turbo decoder interleaver as an iterative process. The output $\vec{x}^e$ from the second constituent decoder is obtained after the decoding operation is completed.

[0020] As discussed above, the use of a turbo code interleaver requires that coding be performed on a finite sequence length. To encode such a finite information sequence, it is necessary for both constituent RSC encoders in the turbo encoder to start and end in an all zero-state with the same trellis bits. Most prior art turbo encoders have their information sequences terminated with a plurality of tail bits. Tail bits are considered a nuisance and as overhead of the turbo encoded sequence.

[0021] The difficulties with flushing turbo code encoders and bringing their trellises back to their initial state have long been recognized by the prior art. For example, the article entitled Turbo Code Termination And Interleaver Conditions by Blackert et al., the article entitled Turbo Codes For PSC Applications by Divsalar et al., and the article entitled Terminating The Trellis Of Turbo-Codes In The Same State by Barbulescu et al. recognize the problems inherent in bringing the trellises of multiple encoders back to their initial states. However, none of these prior art solutions provide a suitable method for bringing the trellises of multiple encoders back to their initial state without reduction in the efficiency of the encoder.

[0022] Accordingly, there exists a need for a turbo code interleaver that does not require a plurality of tail bits to force each constituent encoder to an all-zero state.

[0023] SUMMARY OF THE INVENTION

[0024] The present invention relates to a turbo code hybrid interleaver having recursive systematic constituent encoders. The system and process encodes a finite frame of bits without requiring a plurality of tail bits to flush the registers of each encoder to an all-zero state. The hybrid interleaver reduces the turbo code overhead by using the same tail bits for both constituent encoders improving the performance of the best turbo interleaver.

[0025] Accordingly, it is an object of the present invention to provide a system and method of interleaving that does not require a plurality tail bits to be part of the encoding process.

[0026] It is a further object of the invention to eliminate the unnecessary overhead in the turbo code encoding sequence limiting the number of tail bits that terminate the encoding process to an all-zero state with a single m-bit tail where m is the number of storage registers in each constituent encoder.

[0027] Other objects and advantages of the system and the method will become apparent to those skilled in the art after reading the detailed description of the preferred embodiment.

[0028] BRIEF DESCRIPTION OF THE DRAWINGS

[0029] Figure 1 is a prior art trellis diagram for a 4 state RSC encoder.

[0030] Figure 2 is a system diagram of a prior art, turbo code encoder.

[0031] Figure 3 is a system diagram of a prior art, four state encoder showing tailing off.

[0032] Figure 4 is a flowchart of a prior art method of tailing off.

[0033] Figure 5 is a system diagram of a prior art, turbo code decoder.

[0034] Figure 6 is a system diagram of a turbo code encoder with a hybrid interleaver employing the system and method of the present invention.

[0035] Figure 7 is a flowchart of the interleaver method embodying the present invention.

[0036] Figure 8 is a 16 frame size interleaving sequence produced by the present invention for a 4 state turbo code encoder with S equal to 2 and L equal to 4.

[0037] Figure 9 is the mapping of the interleaving sequence of Figure 8.

[0038] Figure 10 is the 16 frame size interleaving sequence of Figure 8 verified.

[0039] Figure 11 is a flowchart of the tailing off method embodying the present invention.

[0040] Figure 12 is a flowchart of an alternative embodiment.

[0041]DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0042] A turbo code encoder 17 with a hybrid interleaver 19 taught in accordance with the present invention as shown in Figure 6 terminates the first 21 and second 23 RCS constituent encoders to an all-zero state using a single tailing off bit operation 25. The present invention 17 exploits the cyclic property of each constituent encoder 21, 23 in conjunction with keeping the performance of the best turbo interleavers. The turbo code encoder 17 with hybrid interleaver 19 reduces additional tail bit overhead necessary for trellis termination of each constituent encoder 21, 23.

[0043] Figures 6 and 7, describe the system and process of the hybrid turbo code interleaver 19. The process 51 begins (step 53) by receiving a sequence of input data 27 for encoding. The encoding sequence frame size *N* is chosen (step 55). The state size and puncturing rate (code rate) are independent of the hybrid interleaver 19. The hybrid interleaver 19 generates the random integers *I(k)* for permutation (step 57).

[0044] As shown in Figures 8 and 9, the generation of the random integer sequence is performed bit by bit for each frame 29 position $31_{1\text{-}N}$. The generation of a random integer (step 57) denoted as *I(k)* is:

$$1 \leq I(k) \leq N \qquad \text{Equation (1)}$$

where $k = 1,2,\ldots,N$ for each mapped 33 position $35_{1\text{-}N}$ in the interleaver sequence. The current selection, *I(k)* must meet conditions *A* (step 59), *B* (step 63) and *C* (step 65) as follows.

$$Condition\ A\colon \left|I(k) - I\left(k - j\right)\right| > S \qquad \text{Equation (2)}$$

where

$$0 < j \leq S \qquad \text{Equation (3)}$$

and

$$k - j \geq 0. \qquad \text{Equation (4)}$$

Condition A Equation (2) represents the properties of *S*-random interleavers. S is an arbitrary value.

$$Condition\ B\colon \left|I(k) - I(k - n \cdot L)\right| \neq j \cdot L \qquad \text{Equation (5)}$$

(step 63) where n and j are positive integers subject to:

$$k - n \cdot L \geq 0; \qquad \text{Equation (6)}$$

; and

$$n \cdot L \leq S \qquad \text{Equation (7)}$$

(step 61)

L is determined by the constituent encoder used in the turbo code encoder. As an example, *L*=7 is used in an eight state turbo encoder.

$$\textit{Condition C}: k \bmod 2^m - 1 = I(k) \bmod 2^m - 1 \; \forall \, k$$

Equation (9)

(step 65) where m is the size of memory in the constituent encoder. For 4 and 8 state encoders, m equals 2 and 3 respectively. The above steps are repeated until all of the integers, *I(k)* for k=1,2,...,*N*, (step 66) for the hybrid interleaver 19 are selected (step 67) and output (step 69).

[0045] An example of the above system and method is shown in Figures 8, 9 and 10. A sequence frame size of 16 using a 4 state turbo code encoder 17 with hybrid interleaver 19 with *S* equal to 2 and *L* equal to 4 is shown permuted in accordance with the teachings of the invention. The hybrid interleaver 19 satisfies *Conditions A* and *B*. The hybrid interleaver 19 output 37 is verified in Figure 10 using *Condition C* such that after dividing the index of an input 27 information sequence by $2^m - 1$, the resulting remainder sequence 39A is equal to the corresponding remainder sequence 39B due to the interleaving mapping index 33. Once the turbo code hybrid interleaver 19 is specified 51, the information bits 27 are permuted according the hybrid interleaver 19 in order for the second 23 constituent encoder to receive the output 37.

[0046] The process of the present invention that terminates the trellis using the same tail bits for the first 21 and second 23 constituent encoders is shown and described in Figures 6 and 11. As described above, the information bits are encoded by both encoders. The first 21 constituent encoder operates on the information bits 27 in their original order.

The second 23 constituent encoder operates on the information bits 27 as permuted 37 according to the hybrid interleaver 19. The output from the first 21 and second 23 constituent encoders are punctured and multiplexed producing an output (see Figure 2).

[0047] The trellis termination process 81 using the same tail bits for both constituent encoders starts (step 83) with acknowledging that all of the information bits have been encoded by the first 21 and second 23 constituent encoders. At this time in the encoding process, the register contents of both encoders are the same. The first 21 and second 23 encoders switch inputs from the original information 27 and permuted 37 bit streams to feedback 41 from the first 21 encoder. The puncturing of the first 21 encoder output $\vec{p}^{1}{}_{N}$ and the second 23 output $\vec{p}^{2}{}_{N}$ with the information output $\vec{x}_N$ for the tailing off process is the same as during the encoding 21, 23 of the information bits 27, 37. After both switches 43, 45 transition, the first 21 encoder receives tail bits from its own register via the feedback 41 (step 85). The tail bits to the second 23 encoder have not been interleaved by the hybrid interleaver 19 and are the same tail bits 41 for trellis termination as in the first 21 encoder (step 87).

[0048] For a M state encoder, $\log_2$ M tail bits are required to flush all of the registers in the first 21 and second 23 encoders to an all-zero state. With $L = \log_2$ M, Table 1 shows the required number of tail bits and the total number of tail coded symbols for a 4 and 8 state encoder.

		L	Total coded bits at tail part (prior art)	Total coded bits at tail part (present invention)
8-state encoder	½ rate Turbo code	3	2 X 6 = 12	6
	⅓ rate Turbo code	3	2 X 9 = 18	9

4-state encoder	½ rate Turbo code	2	2 X 4 = 8	4
	⅓ rate Turbo code	2	2 X 6 = 12	6

TABLE 1

[0049] For a ½ rate and 1/3 rate turbo code encoder with four (4) state constituent encoders, the present invention 17 eliminates 4 and 6 tail bits, respectively. For a ½ rate and 1/3 rate turbo code encoder with eight (8) state constituent encoders, the present invention 17 eliminates 6 and 9 tail bits, respectively, as compared to and required by the prior art.

[0050] The turbo code encoder with the hybrid interleaver yields better performance than prior art S-random interleavers since the rules stated in *Condition B* avoids worst case low weight distribution of the turbo codes while *Condition A* retains the best characteristics. Since the hybrid interleaver 19 leads to the same trellis state sequences for both the first 21 and second 23 constituent decoders at the beginning of the tail part, the use of a single m-bit tail sequence to flush both the first 21 and second 23 encoders to an all-zero state is acceptable. The extrinsic information $\vec{L}^{I}{}_{e1}$ including tail bits generated from the first constituent decoder are passed on to the second constituent decoder which increases to overall performance (see Figure 5). As an example, if the original information sequence is

$$\vec{x}_N = \{1\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 1\}.$$

The permuted information sequence according to the hybrid interleaver 19 is

$$\vec{x}^{I}{}_{N} = \{0\ 0\ 0\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 1\}.$$

[0051] The information sequence is encoded by the first 21 and second 23 constituent encoders. The first 21 constituent encoder operates on the input $\vec{x}$ in its original order, while the second 23 constituent encoder operates on the permuted $\vec{x}^{I}$ interleaver 19 output.

[0052] The trellis state sequence obtained from the first 21 encoder is

$$\{2\ 3\ 3\ 3\ 1\ 0\ 0\ 0\ 0\ 2\ 1\ 0\ 0\ 2\ 3\ 3\ 1\ 0\}.$$

The trellis state sequence obtained from the second 23 encoder is

$$\{0\ 0\ 0\ 2\ 3\ 3\ 3\ 3\ 3\ 1\ 0\ 0\ 2\ 3\ 3\ 3\ 1\ 0\}.$$

[0053] As shown above, the last two states (four bits) from each trellis state sequence are the same due to the hybrid interleaver 19. This allows the first 21 and the second 23 constituent encoders to receive the same tail bits leading to the reduced overhead of the turbo coding process.

[0054] Condition C leads the trellis state of two constituent encoders to be the same after encoding information bits. This allows the same tail bits for both constituent encoders, resulting in the reduction of turbo-code overhead due to tail bits. In addition, using the same tail bits is desirable for an interative decoder as previously explained in which the interleaver design was based on a S-random interleaver. While the present invention improves turbo-code performance, its memory requirement is the same as for the S-random interleaver with the memory storage requirement proportional to the interleaver size.

[0055] An alternative embodiment is described in Figure 12.

[0056] Let D denote the information sequence of binary bits with block size N such that:

$$D = \{d_{1,}d_{2,}\ldots,d_N\} \textit{ where } d_k = \pm 1 \qquad \text{Equation (10)}$$

[0057] Given a M-state turbo-coder where M is equal to 4 or 8, we can partition the information sequence, D, into p-disjoint subsets, S, where p=M-1 as follows:

$$So = \{d_k|, k \bmod p = 0\} \qquad \text{Equation (11)}$$

$$S_1 = \{d_k|, k \bmod p = 1\} \qquad \text{Equation (12)}$$

$$\vdots$$

$$S_{p-1} = \{d_k|, k \bmod p = p-1\}$$

Equation (13)

where p is set to be 3 and 7 for 4-state and 8-state turbo codes, respectively. The above partition method is similar to the above coset partitioning. The value of p for each state Turbo-code is specified.

[0058] Each subset has the block size of $\lfloor N/p \rfloor$ where $\lfloor N/p \rfloor$ denotes the smallest integer value larger than or equal to *N/p*. Each subset is permuted by the use of any interleaver mapping. Then we combine all the individual subsets in order to obtain the entire interleaver output, denoted as *I*, as follows:

```
Count=$0$;
for $k=1$: Block _size of subset
        for $i= 1$: P
          if i=p
   $I$(count)= $S_0(k)$
          else
            $I$(count)= $S_i(k)$
          end if
  count = count + 1
       if count =$N$
   exit
        end
end
```

where $S_i(k)$ is the k^{th} interleaved output bit of the subset S_i and S_0 (k) is the k^{th} interleaved output bit of the subset S_0. The above mentioned procedures including partition and combining subsets can be re-illustrated by using a block interleaver with $\lfloor N/p \rfloor$ rows and p columns as follows:

1) The information bits are stored row-wise in the block interleaver as follows:

Write

d_1	d_2	d_3	.....	d_p
d_{p+1}	d_{p+2}	d_{p+3}	.....	d_{2p}
d_{2p+1}	d_{2p+2}	d_{2p+3}	.....	d_{3p}
d_{3p+1}	d_{3p+2}	d_{3p+3}	.....	d_{4p}
:	:	:	.....	:
$d_{N/p+1}$	$d_{N/p+2}$	$d_{N/p+3}$	.....	$d_{N/p+p}$

⇑ ⇑ ⇑ ⇑

S_1 S_2 S_3 S_0

2) Permute the bits within each column block according to the given interleaver type, which can be, in principle, one of any candidate interleavers. For example, applying conditions A and B to each column block; condition C is not necessary under these circumstances.

3) Read out the matrix row-by-row in order as shown below to drive the second constituent, whose input is the interleaved output sequence, to the same state as without interleaving the original information sequence.

Read

$d_{N/p}+1$	$d_{N/p+2}$	$d_{N/p+3}$	....	$d_{N/p+p}$
d_{3p+1}	d_{3p+2}	d_{3p+3}	....	d_{4p}
d_{2p+1}	d_{2p+2}	d_{2p+3}	....	d_{3p}
:	:	:	....	:
d_{p+1}	d_{p+2}	d_{p+3}	....	d_{2p}
d_1	d_2	d_3	....	d_p

⇑ ⇑ ⇑ ⇑

S_1 S_2 S_3 S_0

[0058] While the present invention has been described in terms of the preferred embodiment, other variations which are in the scope of the invention as outlined in the claims below will be apparent to those skilled in the art.

* * *